Compressive Sensing

Dr.-Ing. Ali Bereyhi

Friedrich-Alexander University

ali.bereyhi@fau.de

Summer 2022

Where are We?

- *Optimal recovery is given by ℓ_0-norm minimization*
- *Best relaxation of optimal approach is ℓ_1-norm minimization*
  - *We learned various forms of ℓ_1-norm minimization*
  - *We also learned how to implement it*
- *We learned various iterative algorithms as well*
  - *Greedy algorithms: OMP, CoSaMP and Subspace Pursuit*
  - *Thresholding algorithms: IHT, HTP and IST*

Where are We?

Up to this point, we have assumed noiseless sampling, i.e.,

Samples are modeled as $\boldsymbol{y} = \mathbf{A}\boldsymbol{x}$

However, in practice we usually have noisy sampling

Samples are modeled as $\boldsymbol{y} = \mathbf{A}\boldsymbol{x} + \boldsymbol{w}$

We now want to modify the sparse recovery algorithms,

In order to deal with sampling noise

Noisy Sparse Recovery

Noisy Signal Sampling

Remember the signal model

$$x(t) = \sum_{n=1}^{N} x_n \exp\{2\pi j f_n t\}$$

The noise-free sample at time instance $t = t_m$ is

$$y_m = x(t_m) = \sum_{n=1}^{N} x_n e^{2\pi j f_n t_m} = \underbrace{\left[e^{2\pi j f_1 t_m}, \ldots, e^{2\pi j f_N t_m} \right]}_{\boldsymbol{a}_m^{\mathsf{T}}} \underbrace{\begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}}_{\boldsymbol{x}}$$

$$= \boldsymbol{a}_m^{\mathsf{T}} \boldsymbol{x}$$

Noisy Signal Sampling

In practice, sampling is noisy, i.e., we observe

$$y_m = \mathbf{a}_m^{\mathsf{T}} \mathbf{x} + w_m$$

What is w_m?

It is an unknown number with following properties:

- w_m *is positive or negative equal chance*
- *Almost all the time* $|w_m| \leq \epsilon$

Noisy Signal Sampling

In practice, sampling is noisy, i.e., we observe

$$y_m = \boldsymbol{a}_m^{\mathsf{T}} \boldsymbol{x} + w_m$$

So, how we could model it?

We could model it as a random process

- w_m *has zero mean*
- *It has a limited variance*

This is enough for now! We later consider a more specific model

Sparse Recovery from Noisy Samples

Sparse Recovery

$$\mathbf{A}\boldsymbol{z} = \boldsymbol{y} \qquad \textit{subject to } \|\boldsymbol{z}\|_0 \leq s$$

How does the sparse recovery problem look like now?

Even for the true signal $\boldsymbol{x}$, we have in this case

$$\boldsymbol{y} \neq \mathbf{A}\boldsymbol{x}$$

So, clearly we cannot use the earlier formulation!

Sparse Recovery from Noisy Samples

Sparse Recovery

$$\mathbf{A}\boldsymbol{z} = \boldsymbol{y} \qquad \text{subject to } \|\boldsymbol{z}\|_0 \leq s$$

How does the sparse recovery problem look like now?

We however can say in this case that

$$\|\boldsymbol{y} - \mathbf{A}\boldsymbol{x}\|_2^2 = \sum_{m=1}^{M} \left(y_m - \boldsymbol{a}_m^\mathsf{T} \boldsymbol{x} \right)^2 = \sum_{m=1}^{M} w_m^2$$

Sparse Recovery from Noisy Samples

Sparse Recovery

$$\mathbf{A}\boldsymbol{z} = \boldsymbol{y} \qquad \textit{subject to } \|\boldsymbol{z}\|_0 \leq s$$

How does the sparse recovery problem look like now?

Since $|w_m| \leq \epsilon$, *we could say*

$$\|\boldsymbol{y} - \mathbf{A}\boldsymbol{x}\|_2^2 = \sum_{m=1}^{M} w_m^2 \leq M\epsilon^2$$

Sparse Recovery from Noisy Samples

So, the sparse solution satisfies $\|\boldsymbol{x}\|_0 \leq s$, *and*

$$\|\boldsymbol{y} - \mathbf{A}\boldsymbol{x}\|_2^2 \leq M\epsilon^2$$

We could hence formulate the sparse recovery problem as below:

Noisy Sparse Recovery

For sampling matrix $\mathbf{A}$ *and samples* $\boldsymbol{y}$ *find* $\boldsymbol{z}$, *such that*

$$\|\mathbf{A}\boldsymbol{z} - \boldsymbol{y}\|_2^2 \leq M\epsilon^2 \qquad \text{subject to } \|\boldsymbol{z}\|_0 \leq s$$

Sparse Recovery from Noisy Samples

Noisy Sparse Recovery

For sampling matrix $\mathbf{A}$ *and samples* $\boldsymbol{y}$ *find* $\boldsymbol{z}$*, such that*

$$\|\mathbf{A}\boldsymbol{z} - \boldsymbol{y}\|_2^2 \leq M\epsilon^2 \qquad \textit{subject to } \|\boldsymbol{z}\|_0 \leq s$$

This formulation also includes the earlier form

Set $\epsilon \to 0$*; then, we have*

$$\|\mathbf{A}\boldsymbol{z} - \boldsymbol{y}\|_2^2 \leq 0 \rightsquigarrow \|\mathbf{A}\boldsymbol{z} - \boldsymbol{y}\|_2^2 = 0$$
$$\rightsquigarrow \mathbf{A}\boldsymbol{z} = \boldsymbol{y} \rightsquigarrow \textit{noise-free case}$$

Sparse Recovery from Noisy Samples

Is the solution unique in this case?

Assume that $\boldsymbol{x}$ *is the true signal with* $\text{Supp}(\boldsymbol{x}) = \mathcal{S}$. *We now construct a new signal* $\boldsymbol{x}^\sharp$ *out of* $\boldsymbol{x}$ *as below:*

- *Choose* $j \in \mathcal{S}$
- *Set* $x_j^\sharp = x_j + \alpha$ *and* $x_n^\sharp = x_n$ *for or* $n \neq j$

We could hence write

$$\begin{aligned} \|\mathbf{A}\boldsymbol{x}^\sharp - \boldsymbol{y}\|_2^2 &= \|\mathbf{A}\boldsymbol{x} - \boldsymbol{y} + \alpha \boldsymbol{a}_j\|_2^2 \\ &\leq \|\mathbf{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \|\alpha \boldsymbol{a}_j\|_2^2 \\ &= \|\mathbf{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \alpha^2 \|\boldsymbol{a}_j\|_2^2 \end{aligned}$$

Sparse Recovery from Noisy Samples

Is the solution unique in this case?

Let columns of $\mathbf{A}$ *be normalized and remember* $\boldsymbol{w} = \mathbf{A}\boldsymbol{x} - \boldsymbol{y}$

$$\|\mathbf{A}\boldsymbol{x}^\sharp - \boldsymbol{y}\|_2^2 \leq \|\boldsymbol{w}\|_2^2 + \alpha^2$$

This means that if $\|\boldsymbol{w}\|_2^2 < M\epsilon^2$, *for any* α *with*

$$0 < \alpha^2 \leq M\epsilon^2 - \|\boldsymbol{w}\|_2^2$$

Thus, $\boldsymbol{x}^\sharp$ *is also a solution to the sparse recovery problem!*

Sparse Recovery from Noisy Samples

Is the solution unique in this case?

What is the chance of having $\|\boldsymbol{w}\|_2^2 < M\epsilon^2$, i.e.,

$$\Pr\left\{\|\boldsymbol{w}\|_2^2 \neq M\epsilon^2\right\}$$

Well! It depends on distribution of $\boldsymbol{w}$, *but for continuous* $\boldsymbol{w}$

$$\Pr\left\{\|\boldsymbol{w}\|_2^2 \neq M\epsilon^2\right\} = 1$$

This means that for almost all realizations,

The solution is not unique

Sparse Recovery from Noisy Samples

If the solution is not unique, which one should be taken?

In fact, in this case, we are not recovering the true signal

We are estimating it by an estimate

The choice of this estimate however depends on

The metric by which we evaluate the quality

For now, we focus on the LS method

We get back to this point later on in the next chapters

Sparse Recovery via LS Method

Residual Sum of Squares

One metric to evaluate the quality of our estimate is

Residual Sum of Squares (RSS)

Residual Sum of Squares

Let $\boldsymbol{z}$ with $\|\boldsymbol{z}\|_0 \leq s$ be an estimate of the true signal $\boldsymbol{x}$. The RSS for this estimate is defined as

$$\Delta(\boldsymbol{z}) = \|\mathbf{A}\boldsymbol{z} - \boldsymbol{y}\|_2^2$$

Why should RSS be a good metric?

We need few definitions to answer this question

Residual Sum of Squares

Singular Values

For matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$*, the singular values are defined as*

$$\sigma_m(\mathbf{A}) = \sqrt{\lambda_m(\mathbf{A}\mathbf{A}^\mathsf{T})}$$

for $m \in \{1, \dots, M\}$*, where* $\lambda_m\left(\mathbf{A}\mathbf{A}^\mathsf{T}\right)$ *is m-th eigenvalue of* $\mathbf{A}\mathbf{A}^\mathsf{T}$

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 3 \\ 2 & 1 & 1 \end{bmatrix} \rightsquigarrow \mathbf{A}\mathbf{A}^\mathsf{T} = \begin{bmatrix} 11 & 6 \\ 6 & 6 \end{bmatrix} \rightsquigarrow \begin{cases} \lambda_1 = 2 \\ \lambda_2 = 15 \end{cases}$$

The singular values are hence $\sigma_1 = \sqrt{2}$ *and* $\sigma_2 = \sqrt{15}$

Residual Sum of Squares

Singular values of a matrix bound the norm of its projections

Bounding Norm of Projections

For any $\boldsymbol{x} \in \mathbb{R}^N$ and matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$, we have

$$\sigma_{\min}^2(\mathbf{A}) \|\boldsymbol{x}\|_2^2 \leq \|\mathbf{A}\boldsymbol{x}\|_2^2 \leq \sigma_{\max}^2(\mathbf{A}) \|\boldsymbol{x}\|_2^2$$

and

$$\sigma_{\min}(\mathbf{A}) = \min_m \sigma_m(\mathbf{A}) \qquad \text{and} \qquad \sigma_{\max}(\mathbf{A}) = \max_m \sigma_m(\mathbf{A})$$

You prove these bounds using singular value decomposition of $\mathbf{A}$

Residual Sum of Squares

This result can be represented alternatively as

Bounding Norm of Projections

For any $\mathbf{x} \in \mathbb{R}^N$ *and matrix* $\mathbf{A} \in \mathbb{R}^{M \times N}$, *there exist a* κ

$$\sigma_{\min}^2 (\mathbf{A}) \leq \kappa \leq \sigma_{\max}^2 (\mathbf{A})$$

for which we have

$$\|\mathbf{A}\mathbf{x}\|_2^2 = \kappa \|\mathbf{x}\|_2^2$$

Residual Sum of Squares

Why should RSS be a good metric?

Let's assume $\boldsymbol{w}$ *is a random vector whose entries have*

zero mean and variance ξ^2

This means that for a vector $\boldsymbol{a} \in \mathbb{R}^M$*, we have*

$$\mathcal{E}\left[\boldsymbol{a}^\mathsf{T}\boldsymbol{w}\right] = \mathcal{E}\left[\sum_{m=1}^{M} a_m w_m\right] = \sum_{m=1}^{M} a_m \mathcal{E}\left[w_m\right] = 0$$

$$\mathcal{E}\left[\|\boldsymbol{w}\|_2^2\right] = \mathcal{E}\left[\sum_{m=1}^{M} |w_m|^2\right] = \sum_{m=1}^{M} \mathcal{E}\left[|w_m|^2\right] = M\xi^2$$

Residual Sum of Squares

Why should RSS be a good metric?

The average RSS is then given by

$$
\begin{aligned}
\mathcal{E}\left[\Delta\left(\boldsymbol{z}\right)\right] &= \mathcal{E}\left[\|\mathbf{A}\boldsymbol{z} - \boldsymbol{y}\|_2^2\right] \\
&= \mathcal{E}\left[\|\mathbf{A}\boldsymbol{z} - \mathbf{A}\boldsymbol{x} - \boldsymbol{w}\|_2^2\right] \\
&= \mathcal{E}\left[\|\mathbf{A}\left(\boldsymbol{z} - \boldsymbol{x}\right) - \boldsymbol{w}\|_2^2\right] \\
&= \|\mathbf{A}\left(\boldsymbol{z} - \boldsymbol{x}\right)\|_2^2 - 2\underbrace{\mathcal{E}\left[\left(\boldsymbol{z} - \boldsymbol{x}\right)^{\mathsf{T}}\mathbf{A}^{\mathsf{T}}\boldsymbol{w}\right]}_{0} + \underbrace{\mathcal{E}\left[\|\boldsymbol{w}\|_2^2\right]}_{M\xi^2} \\
&= \|\mathbf{A}\left(\boldsymbol{z} - \boldsymbol{x}\right)\|_2^2 + M\xi^2
\end{aligned}
$$

Residual Sum of Squares

Why should RSS be a good metric?

Using, the bound on the projection norm, we could write

$$\begin{aligned}\mathcal{E}\left[\Delta\left(\boldsymbol{z}\right)\right] &= \left\|\mathbf{A}\left(\boldsymbol{z}-\boldsymbol{x}\right)\right\|_2^2 + M\xi^2\\ &= \kappa\|\boldsymbol{z}-\boldsymbol{x}\|_2^2 + M\xi^2\end{aligned}$$

for a κ bounded by singular values of $\mathbf{A}$. We could hence say

$$\textit{Average RSS} \propto \|\boldsymbol{z}-\boldsymbol{x}\|_2^2$$

Thus, bounding RSS leads to restriction of the recovery error

Sparse Recovery via LS Method

Section 1: Regularizing LS via ℓ_0-Norm

Regularizing LS for Sparse Recovery

The LS method finds an estimate of the signal by minimizing RSS

This could however lead to non-sparse solution

To restrict the LS method to sparse recovery,

we regularize LS via the sparsity constraint

LS Regularized via ℓ_0-Norm

$$\min_{\boldsymbol{z}} \|\mathbf{A}\boldsymbol{z} - \boldsymbol{y}\|_2^2 \qquad \textit{subject to} \quad \|\boldsymbol{z}\|_0 \leq s$$

Regularizing LS for Sparse Recovery

LS Regularized via ℓ_0-Norm

$$\min_{\boldsymbol{z}} \|\mathbf{A}\boldsymbol{z} - \boldsymbol{y}\|_2^2 \qquad \textit{subject to} \ \ \|\boldsymbol{z}\|_0 \leq s$$

How could we get rid of s?

We could find an equivalent optimization

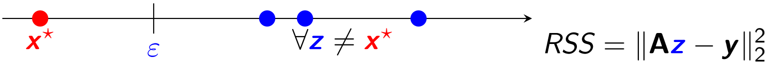
Constraint $\leftrightsquigarrow$ *Objective*

We have done it before in this lecture

Regularized LS: ℓ_0-Norm Minimization

Assume LS regularized via ℓ_0-norm has the unique solution $\boldsymbol{x}^\star$

Let's plot the RSS for all $\boldsymbol{z}$ which satisfy $\|\boldsymbol{z}\|_0 \leq s$



- $\boldsymbol{x}^\star$ *is the unique solution*
- $\boldsymbol{z}$ *are all other points with* $\|\boldsymbol{z}\|_0 \leq s$

We could now think of one ε, for which

$$\|\mathbf{A}\boldsymbol{x}^\star - \boldsymbol{y}\|_2^2 \leq \varepsilon \qquad \text{and} \qquad \|\mathbf{A}\boldsymbol{z} - \boldsymbol{y}\|_2^2 > \varepsilon$$

Regularized LS: ℓ_0-Norm Minimization

Assume LS regularized via ℓ_0-norm has the unique solution $\mathbf{x}^\star$

Since the solution is unique, we could alternatively say

Among all signals $\mathbf{z}$ which satisfy $\|\mathbf{A}\mathbf{z} - \mathbf{y}\|_2^2 \leq \varepsilon$,

$\mathbf{x}^\star$ is the only one with $\|\mathbf{x}^\star\|_0 \leq s$

This means that any $\mathbf{z} \neq \mathbf{x}^\star$ which satisfies $\|\mathbf{A}\mathbf{z} - \mathbf{y}\|_2^2 \leq \varepsilon$ has

$$\|\mathbf{z}\|_0 > s$$

Regularized LS: ℓ_0-Norm Minimization

Assume LS regularized via ℓ_0-norm has the unique solution $\boldsymbol{x}^\star$

This concludes that

$\boldsymbol{x}^\star$ *has minimum sparsity among all* $\boldsymbol{z}$ *with* $\|\boldsymbol{z}\|_0 \leq \varepsilon$

Modified ℓ_0-Norm Minimization

$$\boldsymbol{x}^\star = \operatorname*{argmin}_{\boldsymbol{z}} \|\boldsymbol{z}\|_0 \qquad \textit{subject to} \quad \|\mathbf{A}\boldsymbol{z} - \boldsymbol{y}\|_2^2 \leq \varepsilon$$

Sparse Recovery via LS Method

Section 2: Regularizing LS via ℓ_1-Norm

LS with ℓ_1-Norm Regularization

$\boxed{\textit{Any form of } \ell_0\textit{-norm minimization is NP-hard}}$

Similar to noise-free case, we address the computational issue by

ℓ_1-norm relaxation

Starting from the modified ℓ_0-norm minimization, we end up with

Basis Pursuit with *Quadratic Constraint*

$$\boldsymbol{x}^\star = \operatorname*{argmin}_{\boldsymbol{z}} \|\boldsymbol{z}\|_1 \qquad \textit{subject to} \;\; \|\mathbf{A}\boldsymbol{z} - \boldsymbol{y}\|_2^2 \leq \varepsilon$$

Unlike the noise-free case, now ε does not tend to zero!

LS with ℓ_1-Norm Regularization

As we showed in Part 2, we could alternatively solve the following:

Basis Pursuit *Denoising*

$$\boldsymbol{x}^\star = \operatorname*{argmin}_{\boldsymbol{z}} \frac{1}{2} \|\mathbf{A}\boldsymbol{z} - \boldsymbol{y}\|_2^2 + \lambda \|\boldsymbol{z}\|_1$$

Again we note that, in noisy scenarios λ does not tend to zero!

Remember that optimal λ varies proportional to ε

$$\lambda \uparrow \rightsquigarrow \varepsilon \uparrow \qquad \lambda \downarrow \rightsquigarrow \varepsilon \downarrow$$

LS with ℓ_1-Norm Regularization

LASSO also describes an alternative form:

LASSO

$$\boldsymbol{x}^\star = \operatorname*{argmin}_{\boldsymbol{z}} \|\mathbf{A}\boldsymbol{z} - \boldsymbol{y}\|_2^2 \qquad \textit{subject to} \quad \|\boldsymbol{z}\|_1 \leq \tau$$

Remember the initial form of regularized LS

LS Regularized via ℓ_0-Norm

$$\min_{\boldsymbol{z}} \|\mathbf{A}\boldsymbol{z} - \boldsymbol{y}\|_2^2 \qquad \textit{subject to} \quad \|\boldsymbol{z}\|_0 \leq s$$

So, here τ is proportional to the sparsity

LS with ℓ_1-Norm Regularization

LASSO also describes an alternative form:

LASSO

$$\boldsymbol{x}^\star = \operatorname*{argmin}_{\boldsymbol{z}} \|\mathbf{A}\boldsymbol{z} - \boldsymbol{y}\|_2^2 \qquad \textit{subject to} \;\; \|\boldsymbol{z}\|_1 \leq \tau$$

One may note that . . .

LASSO is in fact the original form of the LS method which is

regularized by ℓ_1-norm

This is in fact how LASSO was developed initially by Tibshirani!

Summary

In a nutshell, up to this point we have learned that

- *With noisy samples,*

  $$\textit{Exact recovery} \rightsquigarrow \textit{Signal Estimation}$$

- *The optimal approach is not unique anymore*

  It depends on the metric which quantifies estimation quality

- *LS method considers RSS as the metric*
  - *Noisy sparse recovery in this case reduces to*

    modified version of ℓ_0-norm minimization

  - *The relaxed version is given by ℓ_1-norm minimization*

We now try another approach for noisy sparse recovery

Dantzig Approach

Maximum Error Correlation

A less popular, but yet known, metric for quality of our estimate is

Maximum Error Correlation (MEC)

Maximum Error Correlation

Let $\boldsymbol{z}$ with $\|\boldsymbol{z}\|_0 \leq s$ be an estimate of the true signal $\boldsymbol{x}$. The MEC for this estimate is defined as

$$\Delta(\boldsymbol{z}) = \left\|\mathbf{A}^{\mathsf{T}}(\mathbf{A}\boldsymbol{z} - \boldsymbol{y})\right\|_\infty$$

Maximum Error Correlation

MEC was taken initially as a metric for noisy sparse recovery by

Emmanuel Candes and Terence Tao

Why it is then called Dantzig approach?

> *Well! The so-called "father of linear programming"*
>
> *Georg Dantzig*
>
> *passed away in 2005 while*
>
> *Candes and Tao were working on this approach*
>
> *So, they paid tribute to him*

Dantzig Approach: Derivations

But, why should MEC be a good metric?

Say, a genie has told us that

support of the sparse signal is $\mathcal{S}$

We hence want to find an estimation $\mathbf{z}$ whose support is $\mathcal{S}$

Now the question is ...

Is there an alternative way rather than the LS method?

Dantzig Approach: Derivations

Let us stick to our earlier definition of residual

$$\boldsymbol{r} = \mathbf{A}\boldsymbol{x} - \boldsymbol{y}$$

MEC is the ℓ_∞-norm of the inner product

$$\mathbf{A}^\mathsf{T}\boldsymbol{r}$$

But how does the entries of this inner product look like?

Dantzig Approach: Derivations

How does the entries of this inner product look like?

Consider the noisy sampling model

$$\boldsymbol{y} = \mathbf{A}\boldsymbol{x} + \boldsymbol{w}$$

Residual is written as

$$\begin{aligned} \boldsymbol{r} = \mathbf{A}\boldsymbol{z} - \boldsymbol{y} &= \mathbf{A}\boldsymbol{z} - \mathbf{A}\boldsymbol{x} - \boldsymbol{w} \\ &= \mathbf{A}\left(\boldsymbol{z} - \boldsymbol{x}\right) - \boldsymbol{w} \end{aligned}$$

Dantzig Approach: Derivations

How does the entries of this inner product look like?

Therefore, the inner product looks like

$$\mathbf{A}^{\mathsf{T}} \boldsymbol{r} = \mathbf{A}^{\mathsf{T}} \mathbf{A} \left(\boldsymbol{z} - \boldsymbol{x} \right) - \mathbf{A}^{\mathsf{T}} \boldsymbol{w}$$

There are two main components here

- *Estimation error, i.e.,* $\boldsymbol{z} - \boldsymbol{x}$
- *Sampling noise*

Dantzig Approach: Derivations

How does the entries of this inner product look like?

Let's for a moment forget about the sampling noise; then,

- *If the genie has told us the right support; then,*

  we could recover $\boldsymbol{x}$ exactly

  The good estimation $\boldsymbol{z}$ leads to $\mathbf{A}^{\mathsf{T}}\boldsymbol{r} = \mathbf{0}$

- *If the genie has told us the wrong support; then,*

  our estimation is always inexact

  $\mathbf{A}^{\mathsf{T}}\boldsymbol{r}$ *contains always a non-zero entry*

Dantzig Approach: Derivations

How does the entries of this inner product look like?

Let's for a moment forget about the sampling noise; then,

- *If the genie has told us the right support; then,*

$$\|\mathbf{A}^{\mathsf{T}} \boldsymbol{r}\|_\infty = 0$$

- *If the genie has told us the wrong support; then,*

$$\|\mathbf{A}^{\mathsf{T}} \boldsymbol{r}\|_\infty \neq 0$$

Dantzig Approach: Derivations

What does the noise-free case illustrate?

It indicates that

MEC is proportional to the quality of estimation

More precisely, we could say

The most accurate estimation leads to minimum MEC

What about the noisy case?

For noisy sampling, we could extend everything using few tricks

Dantzig Approach: Derivations

Let us start with a simple inequality:

Consider a vector $\boldsymbol{a} \in \mathbb{R}^N$*; then,*

$$\|\boldsymbol{a}\|_2 = \sqrt{\sum_{n=1}^{N} |a_n|^2} \geq \sqrt{\max_n |a_n|^2} = \max_n |a_n| = \|\boldsymbol{a}\|_\infty$$

So the bottom line is that $\|\boldsymbol{a}\|_\infty \leq \|\boldsymbol{a}\|_2$

Dantzig Approach: Derivations

Another simple inequality is

Consider a vector $\mathbf{a} \in \mathbb{R}^N$*; then,*

$$\|\mathbf{a}\|_2 = \sqrt{\sum_{n=1}^{N} |a_n|^2} \leq \sqrt{N \max_n |a_n|^2}$$
$$= \sqrt{N} \max_n |a_n| = \sqrt{N} \|\mathbf{a}\|_\infty$$

So the bottom line is that $\|\mathbf{a}\|_\infty \geq \|\mathbf{a}\|_2 / \sqrt{N}$

Dantzig Approach: Derivations

Also remember this result that

Bounding Norm of Projections

For any $\boldsymbol{x} \in \mathbb{R}^N$ *and matrix* $\mathbf{A} \in \mathbb{R}^{M \times N}$, *there exist a* κ

$$\sigma^2_{\min}(\mathbf{A}) \leq \kappa \leq \sigma^2_{\max}(\mathbf{A})$$

for which we have

$$\|\mathbf{A}\boldsymbol{x}\|_2^2 = \kappa \|\boldsymbol{x}\|_2^2$$

Here, bottom line is $\|\mathbf{A}\boldsymbol{x}\|_2 = \kappa \|\boldsymbol{x}\|_2$ *where* κ *mainly depends on* $\mathbf{A}$

Dantzig Approach: Derivations

What about the noisy case?

Let's bound the MEC in the noisy case from above first

$$\begin{aligned}
\|\mathbf{A}^{\mathsf{T}}\boldsymbol{r}\|_\infty &= \|\mathbf{A}^{\mathsf{T}}\mathbf{A}\left(\boldsymbol{z}-\boldsymbol{x}\right)-\mathbf{A}^{\mathsf{T}}\boldsymbol{w}\|_\infty \\
&\leq \|\mathbf{A}^{\mathsf{T}}\mathbf{A}\left(\boldsymbol{z}-\boldsymbol{x}\right)\|_\infty + \|\mathbf{A}^{\mathsf{T}}\boldsymbol{w}\|_\infty \\
&\leq \|\mathbf{A}^{\mathsf{T}}\mathbf{A}\left(\boldsymbol{z}-\boldsymbol{x}\right)\|_2 + \|\mathbf{A}^{\mathsf{T}}\boldsymbol{w}\|_2 \\
&= \kappa_1\|\boldsymbol{z}-\boldsymbol{x}\|_2 + \kappa_2\|\boldsymbol{w}\|_2
\end{aligned}$$

Dantzig Approach: Derivations

What about the noisy case?

Similarly, we can bound the MEC in the noisy case from below

$$
\begin{aligned}
\|\mathbf{A}^{\mathsf{T}}\boldsymbol{r}\|_\infty &= \|\mathbf{A}^{\mathsf{T}}\mathbf{A}\left(\boldsymbol{z}-\boldsymbol{x}\right)-\mathbf{A}^{\mathsf{T}}\boldsymbol{w}\|_\infty \\
&\geq \|\mathbf{A}^{\mathsf{T}}\mathbf{A}\left(\boldsymbol{z}-\boldsymbol{x}\right)\|_\infty - \|\mathbf{A}^{\mathsf{T}}\boldsymbol{w}\|_\infty \\
&\geq \frac{1}{\sqrt{N}}\|\mathbf{A}^{\mathsf{T}}\mathbf{A}\left(\boldsymbol{z}-\boldsymbol{x}\right)\|_2 - \|\mathbf{A}^{\mathsf{T}}\boldsymbol{w}\|_2 \\
&= \hat{\kappa}_1\|\boldsymbol{z}-\boldsymbol{x}\|_2 - \kappa_2\|\boldsymbol{w}\|_2
\end{aligned}
$$

Dantzig Approach: Derivations

What do the bounds say?

We cannot

- *play with the κ values, as they depend on the matrix*
- *play with $\|\boldsymbol{w}\|_2$ as is given by noise*

So, the key factor in the estimation which change the MEC is

estimation error $\|\boldsymbol{z} - \boldsymbol{x}\|_2$

Dantzig Approach

Section 1: Sparse Recovery with Dantzig Approach

Optimal Dantzig Approach

We can repeat the same approach, as we took with LS

Now, we replace RSS with MEC

Optimal Dantzig Approach

$$\min_{\boldsymbol{z}} \|\mathbf{A}^{\mathsf{T}} (\mathbf{A}\boldsymbol{z} - \boldsymbol{y})\|_{\infty} \qquad \textit{subject to} \ \ \|\boldsymbol{z}\|_0 \leq s$$

Well, we have the same issue as before

We need to get rid of s

We know very well how to do it

Constraint $\leftrightsquigarrow$ *Objective*

Optimal Dantzig Approach

We hence can find some ε, and rewrite

Optimal Dantzig Approach

$$\min_{\boldsymbol{z}} \|\boldsymbol{z}\|_0 \qquad \textit{subject to} \;\; \left\|\mathbf{A}^{\mathsf{T}}\left(\mathbf{A}\boldsymbol{z} - \boldsymbol{y}\right)\right\|_\infty \leq \varepsilon$$

As always, this approach suffers from

Infeasible Computational Complexity

And as always, we deal with it by

ℓ_1-norm relaxation

Dantzig Approach

Section 2: Dantzig Selector

Dantzig Selector

The relaxed version of optimal Dantzig is called Dantzig selector

Dantzig Selector

$$\min_{\boldsymbol{z}} \|\boldsymbol{z}\|_1 \qquad \textit{subject to} \ \ \|\mathbf{A}^{\mathsf{T}}(\mathbf{A}\boldsymbol{z} - \boldsymbol{y})\|_\infty \leq \varepsilon$$

Attention!

- *Although not used, one could consider noiseless case by sending ε to zero*
- *Main reason of choosing ℓ_∞-norm is more complicated*

Summary

In a nutshell, up to this point we have learned that

- *With noisy samples,*

  Exact recovery $\rightsquigarrow$ *Signal Estimation*

- *The optimal approach is not unique anymore*
- *LS method considers RSS as the metric*
- *Dantzig approach considers MEC as the metric*
  - *Optimal Dantzig approach deal with*

    ℓ_0*-norm minimization*

  - *The* ℓ_1*-norm relaxation is called Dantzig selector*

We now go for iterative approaches

Iterative Algorithms

Noisy Sparse Recovery via Iterative Algorithms

Is there any specific change in the iterative approaches?

> *Basically No!*
>
> *Only the stopping criteria is changed*

How do we change that?

> *We consider a quality metric, e.g., RSS or MEC*
>
> *We stop when the metric is less than a threshold level*

Iterative Algorithms

Section 1: Greedy Algorithms

Orthogonal Matching Pursuit

We had

- *Start with* $\mathbf{x}^{(0)} = \mathbf{0}$ *and* $\mathcal{S}^{(0)} = \emptyset$
- *Update the support for* $t \geq 1$ *as*

$$\mathcal{S}^{(t)} = \mathcal{S}^{(t-1)} \cup \operatorname*{argmax}_{n} \left| \left[\mathbf{A}^{\mathsf{T}} \left(\mathbf{y} - \mathbf{A}\mathbf{x}^{(t-1)} \right) \right]_n \right|$$

- *Set* $\mathbf{z} = \mathbf{A}^{\dagger}_{\mathcal{S}^{(t)}} \mathbf{y}$ *and update* $\mathbf{x}^{(t)}$ *as*

$$\mathbf{x}^{(t)}_{\mathcal{S}^{(t)}} = \mathbf{z}$$

- *Stop at iteration* T, *at which* $\mathbf{A}\mathbf{x}^{(T)} = \mathbf{y}$

Orthogonal Matching Pursuit

Now, we have

- *Start with* $\boldsymbol{x}^{(0)} = \mathbf{0}$ *and* $\mathcal{S}^{(0)} = \emptyset$
- *Update the support for* $t \geq 1$ *as*

$$\mathcal{S}^{(t)} = \mathcal{S}^{(t-1)} \cup \operatorname*{argmax}_{n} \left| \left[\mathbf{A}^{\mathsf{T}} \left(\boldsymbol{y} - \mathbf{A}\boldsymbol{x}^{(t-1)} \right) \right]_n \right|$$

- *Set* $\boldsymbol{z} = \mathbf{A}^{\dagger}_{\mathcal{S}^{(t)}} \boldsymbol{y}$ *and update* $\boldsymbol{x}^{(t)}$ *as*

$$\boldsymbol{x}^{(t)}_{\mathcal{S}^{(t)}} = \boldsymbol{z}$$

- *Stop at iteration* T*, at which* $\|\mathbf{A}\boldsymbol{x}^{(T)} - \boldsymbol{y}\|_2^2 \leq \varepsilon$

Orthogonal Matching Pursuit

Or, we have

- *Start with* $\mathbf{x}^{(0)} = \mathbf{0}$ *and* $\mathcal{S}^{(0)} = \emptyset$
- *Update the support for* $t \geq 1$ *as*

$$\mathcal{S}^{(t)} = \mathcal{S}^{(t-1)} \cup \underset{n}{\operatorname{argmax}} \left| \left[\mathbf{A}^{\mathsf{T}} \left(\mathbf{y} - \mathbf{A}\mathbf{x}^{(t-1)} \right) \right]_n \right|$$

- *Set* $\mathbf{z} = \mathbf{A}^{\dagger}_{\mathcal{S}^{(t)}} \mathbf{y}$ *and update* $\mathbf{x}^{(t)}$ *as*

$$\mathbf{x}^{(t)}_{\mathcal{S}^{(t)}} = \mathbf{z}$$

- *Stop at iteration* T*, at which* $\|\mathbf{A}^{\mathsf{T}} \left(\mathbf{A}\mathbf{x}^{(T)} - \mathbf{y} \right)\|_\infty \leq \varepsilon$

CoSaMP Algorithm

For CoSaMP, we had

- *Start with* $\mathbf{x}^{(0)} = \mathbf{0}$ *and* $\mathcal{S}^{(0)} = \emptyset$, *and choose a* K
- *Update the support for* $t \geq 1$ *as*

$$\hat{\mathcal{S}}^{(t)} = \mathcal{S}^{(t-1)} \cup D_K \left(\mathbf{A}^\mathsf{T} \left(\mathbf{y} - \mathbf{A}\mathbf{x}^{(t-1)} \right) \right)$$

- *Set* $\mathbf{z} = \mathbf{A}^\dagger_{\hat{\mathcal{S}}^{(t)}} \mathbf{y}$ *and update* $\mathbf{u}^{(t)}$ *as* $\mathbf{u}^{(t)}_{\hat{\mathcal{S}}^{(t)}} = \mathbf{z}$
- *Update once again* $\mathcal{S}^{(t)} = D_s \left(\mathbf{u}^{(t)} \right)$ *and* $\mathbf{x}^{(t)} = T^\mathsf{H}_s \left(\mathbf{u}^{(t)} \right)$
- *Stop at iteration* T, *at which* $\mathbf{A}\mathbf{x}^{(T)} = \mathbf{y}$

CoSaMP Algorithm

Now, we have

- *Start with* $\boldsymbol{x}^{(0)} = \mathbf{0}$ *and* $\mathcal{S}^{(0)} = \emptyset$, *and choose a* K
- *Update the support for* $t \geq 1$ *as*

$$\hat{\mathcal{S}}^{(t)} = \mathcal{S}^{(t-1)} \cup D_K \left(\mathbf{A}^{\mathsf{T}} \left(\boldsymbol{y} - \mathbf{A}\boldsymbol{x}^{(t-1)} \right) \right)$$

- *Set* $\boldsymbol{z} = \mathbf{A}^{\dagger}_{\hat{\mathcal{S}}^{(t)}} \boldsymbol{y}$ *and update* $\boldsymbol{u}^{(t)}$ *as* $\boldsymbol{u}^{(t)}_{\hat{\mathcal{S}}^{(t)}} = \boldsymbol{z}$
- *Update once again* $\mathcal{S}^{(t)} = D_s \left(\boldsymbol{u}^{(t)} \right)$ *and* $\boldsymbol{x}^{(t)} = T^{\mathrm{H}}_s \left(\boldsymbol{u}^{(t)} \right)$
- *Stop at iteration* T, *at which* $\| \mathbf{A}\boldsymbol{x}^{(T)} - \boldsymbol{y} \|_2^2 \leq \varepsilon$

CoSaMP Algorithm

Or, we have

- *Start with* $\mathbf{x}^{(0)} = \mathbf{0}$ *and* $\mathcal{S}^{(0)} = \emptyset$, *and choose a* K
- *Update the support for* $t \geq 1$ *as*

$$\hat{\mathcal{S}}^{(t)} = \mathcal{S}^{(t-1)} \cup D_K \left(\mathbf{A}^\mathsf{T} \left(\boldsymbol{y} - \mathbf{A} \boldsymbol{x}^{(t-1)} \right) \right)$$

- *Set* $\boldsymbol{z} = \mathbf{A}^{\dagger}_{\hat{\mathcal{S}}^{(t)}} \boldsymbol{y}$ *and update* $\boldsymbol{u}^{(t)}$ *as* $\boldsymbol{u}^{(t)}_{\hat{\mathcal{S}}^{(t)}} = \boldsymbol{z}$
- *Update once again* $\mathcal{S}^{(t)} = D_s \left(\boldsymbol{u}^{(t)} \right)$ *and* $\boldsymbol{x}^{(t)} = T_s^{\mathrm{H}} \left(\boldsymbol{u}^{(t)} \right)$
- *Stop at iteration* T, *at which* $\left\| \mathbf{A}^\mathsf{T} \left(\mathbf{A} \boldsymbol{x}^{(T)} - \boldsymbol{y} \right) \right\|_\infty \leq \varepsilon$

Subspace Pursuit

For subspace pursuit, we had

- *Start with* $\mathbf{x}^{(0)} = \mathbf{0}$ *and* $\mathcal{S}^{(0)} = \emptyset$*, and choose a* K
- *Update initially the support for* $t \geq 1$ *as*

$$\hat{\mathcal{S}}^{(t)} = \mathcal{S}^{(t-1)} \cup D_K \left(\mathbf{A}^\mathsf{T} \left(\mathbf{y} - \mathbf{A}\mathbf{x}^{(t-1)} \right) \right)$$

- *Set* $\mathbf{z} = \mathbf{A}^\dagger_{\hat{\mathcal{S}}^{(t)}} \mathbf{y}$ *and update* $\mathbf{u}^{(t)}$ *as* $\mathbf{u}^{(t)}_{\hat{\mathcal{S}}^{(t)}} = \mathbf{z}$
- *Update* $\mathcal{S}^{(t)} = D_s \left(\mathbf{u}^{(t)} \right)$
- *Set* $\mathbf{z} = \mathbf{A}^\dagger_{\mathcal{S}^{(t)}} \mathbf{y}$ *and update* $\mathbf{x}^{(t)}$ *as* $\mathbf{x}^{(t)}_{\mathcal{S}^{(t)}} = \mathbf{z}$
- *Stop at iteration* T*, at which* $\mathbf{A}\mathbf{x}^{(T)} = \mathbf{y}$

Subspace Pursuit

Now, we have

- *Start with* $\mathbf{x}^{(0)} = \mathbf{0}$ *and* $\mathcal{S}^{(0)} = \emptyset$, *and choose a* K
- *Update initially the support for* $t \geq 1$ *as*

$$\hat{\mathcal{S}}^{(t)} = \mathcal{S}^{(t-1)} \cup D_K \left(\mathbf{A}^{\mathsf{T}} \left(\mathbf{y} - \mathbf{A}\mathbf{x}^{(t-1)} \right) \right)$$

- *Set* $\mathbf{z} = \mathbf{A}^{\dagger}_{\hat{\mathcal{S}}^{(t)}} \mathbf{y}$ *and update* $\mathbf{u}^{(t)}$ *as* $\mathbf{u}^{(t)}_{\hat{\mathcal{S}}^{(t)}} = \mathbf{z}$
- *Update* $\mathcal{S}^{(t)} = D_s \left(\mathbf{u}^{(t)} \right)$
- *Set* $\mathbf{z} = \mathbf{A}^{\dagger}_{\mathcal{S}^{(t)}} \mathbf{y}$ *and update* $\mathbf{x}^{(t)}$ *as* $\mathbf{x}^{(t)}_{\mathcal{S}^{(t)}} = \mathbf{z}$
- *Stop at iteration* T, *at which* $\|\mathbf{A}\mathbf{x}^{(T)} - \mathbf{y}\|_2^2 \leq \varepsilon$

Subspace Pursuit

Or, we have

- *Start with* $\mathbf{x}^{(0)} = \mathbf{0}$ *and* $\mathcal{S}^{(0)} = \emptyset$, *and choose a* K
- *Update initially the support for* $t \geq 1$ *as*

$$\hat{\mathcal{S}}^{(t)} = \mathcal{S}^{(t-1)} \cup D_K \left(\mathbf{A}^\mathsf{T} \left(\mathbf{y} - \mathbf{A}\mathbf{x}^{(t-1)} \right) \right)$$

- *Set* $\mathbf{z} = \mathbf{A}^\dagger_{\hat{\mathcal{S}}^{(t)}} \mathbf{y}$ *and update* $\mathbf{u}^{(t)}$ *as* $\mathbf{u}^{(t)}_{\hat{\mathcal{S}}^{(t)}} = \mathbf{z}$
- *Update* $\mathcal{S}^{(t)} = D_s \left(\mathbf{u}^{(t)} \right)$
- *Set* $\mathbf{z} = \mathbf{A}^\dagger_{\mathcal{S}^{(t)}} \mathbf{y}$ *and update* $\mathbf{x}^{(t)}$ *as* $\mathbf{x}^{(t)}_{\mathcal{S}^{(t)}} = \mathbf{z}$
- *Stop at iteration* T, *at which* $\| \mathbf{A}^\mathsf{T} \left(\mathbf{A}\mathbf{x}^{(T)} - \mathbf{y} \right) \|_\infty \leq \varepsilon$

Iterative Algorithms

Section 2: Thresholding Algorithms

Iterative Hard Thresholding

Similarly for IHT, we could modify the algorithm as

- *It starts with* $\boldsymbol{x}^{(0)} = \mathbf{0}$ *and* $\mathcal{S}^{(0)} = \emptyset$
- *It updates the approximation of the signal as*

$$\boldsymbol{x}^{(t)} = T_s^{\mathrm{H}} \left(\boldsymbol{x}^{(t-1)} + \mathbf{A}^{\mathsf{T}} \left(\boldsymbol{y} - \mathbf{A}\boldsymbol{x}^{(t-1)} \right) \right)$$

- *It stops at iteration* T*, at which* $\| \mathbf{A}\boldsymbol{x}^{(T)} - \boldsymbol{y} \|_2^2 \leq \varepsilon$

Or alternatively,

- *It stops at iteration* T*, at which* $\| \mathbf{A}^{\mathsf{T}} \left(\mathbf{A}\boldsymbol{x}^{(T)} - \boldsymbol{y} \right) \|_\infty \leq \varepsilon$

Hard Thresholding Pursuit

Similarly, we could modify HTP as

- *It starts with* $\boldsymbol{x}^{(0)} = \mathbf{0}$ *and* $\mathcal{S}^{(0)} = \emptyset$
- *It updates the approximation of the support as*

$$\mathcal{S}^{(t)} = D_s\left(\boldsymbol{x}^{(t-1)} + \mathbf{A}^{\mathsf{T}}\left(\boldsymbol{y} - \mathbf{A}\boldsymbol{x}^{(t-1)}\right)\right)$$

- *It sets* $\boldsymbol{z} = \mathbf{A}^{\dagger}_{\mathcal{S}^{(t)}}\boldsymbol{y}$ *and updates* $\boldsymbol{x}^{(t)}_{\mathcal{S}^{(t)}} = \boldsymbol{z}$
- *It stops at iteration* T, *at which* $\|\mathbf{A}\boldsymbol{x}^{(T)} - \boldsymbol{y}\|_2^2 \leq \varepsilon$

Or we could use MEC and say

- *It stops at iteration* T, *at which* $\|\mathbf{A}^{\mathsf{T}}\left(\mathbf{A}\boldsymbol{x}^{(T)} - \boldsymbol{y}\right)\|_{\infty} \leq \varepsilon$

Iterative Soft Thresholding

Finally, we modify IST as

- *It starts with* $\boldsymbol{x}^{(0)} = \mathbf{0}$ *and* $\mathcal{S}^{(0)} = \emptyset$ *and a fixed* β
- *It updates the approximation of the signal as*

$$\boldsymbol{x}^{(t)} = T_{\lambda}^{\mathsf{S}} \left(\boldsymbol{x}^{(t-1)} + \beta \mathbf{A}^{\mathsf{T}} \left(\boldsymbol{y} - \mathbf{A} \boldsymbol{x}^{(t-1)} \right) \right)$$

- *It stops at iteration* T, *at which* $\|\mathbf{A}\boldsymbol{x}^{(T)} - \boldsymbol{y}\|_2^2 \leq \varepsilon$

Or alternatively,

- *It stops at iteration* T, *at which* $\|\mathbf{A}^{\mathsf{T}} \left(\mathbf{A}\boldsymbol{x}^{(T)} - \boldsymbol{y} \right)\|_\infty \leq \varepsilon$

Final Points

Summary

In a nutshell, we have learned that

- *With noisy samples,*

  Exact recovery $\rightsquigarrow$ *Signal Estimation*

- *The optimal approach is not unique anymore*
- *LS method considers RSS as the metric*
- *Dantzig approach considers MEC as the metric*
- *Iterative algorithms remain the same*

  Only the stop criteria changes: RSS or MEC

What We Learn Next?

We now know most important sparse recovery algorithms, but

$\boxed{\textit{How do they perform?}}$

Answering to this question is quite hard and long!

We go through this long way in the next chapter

Which Parts of Textbooks?

I would suggest to go over

> *Statistical Mechanics of Regularized Least Squares*
> *A. Bereyhi, 2020*

and study the following parts:

- *Chapter 1: Till end of Section 1.3 and Chapter 2: Section 2.3*

You can find the manuscript online at this link

For more details on Dantzig approach, you could check out

> *The Dantzig Selector: Statistical estimation when p is much larger than n, E. Candes and T. Tao, 2005*

You can find the manuscript online at this link